مقدمة إلى المعلوماتية

تكنولوجيا المعلومات

INFORMATION TECHNOLOGY (IT)

تعرف تكنولوجيا المعلومات على أنها استخدام الحواسيب لاستقبال، نقل ومعالجة و تخزين البيانات أو المعلومات. تُستخدم تكنولوجيا المعلومات في العديد من المجالات، وتُعتبر فرعًا من تكنولوجيا المعلومات والاتصالات information and communications technology (ICT).

تصنيف أدوات تكنولوجيا المعلومات والاتصالات (ICT tools) إلى ثلاثة أصناف:

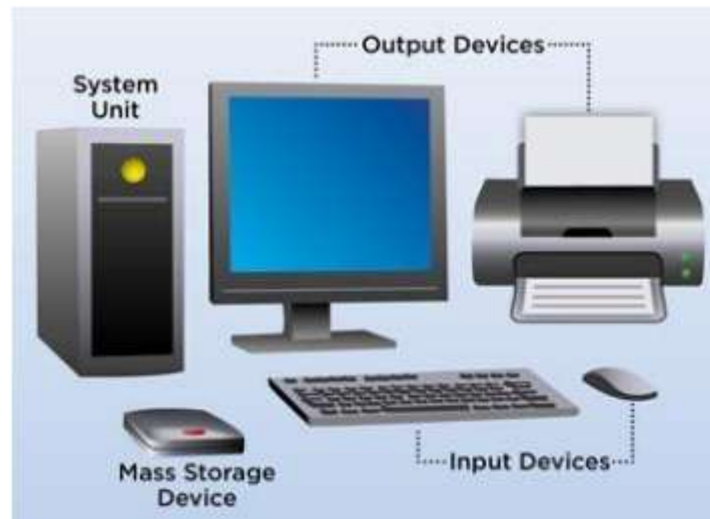
1- **العتاد الصلب (الأجهزة المادية) Physical ICT hardware:** الحواسيب، الهواتف....

2- **العتاد المرن (البرمجيات) ICT software:** البرمجيات، الشبكات اللاسلكية

3- **خدمات تكنولوجيا المعلومات و الاتصالات ICT services:** تعد خدمات الإنترنت والبريد الإلكتروني

نظام الحاسب computer system

هو مجموعة من الأجهزة المتكاملة التي تُدخل البيانات والمعلومات وتعالجها وتُخرجها وتُخزنها. هناك خمسة مكونات رئيسية في نظام الحاسب: أجهزة الإدخال والمعالجة والتخزين والإخراج والاتصالات.



البرمجيات Software

البرمجيات هي مجموعة من التعليمات أو البيانات أو البرامج المستخدمة لتشغيل جهاز الحاسب وتنفيذ مهام محددة. على عكس العتاد، الذي يصف الجوانب المادية لجهاز الحاسب.



البيانات Data

البيانات هي مجموعة من قيم الكائنات المتعلقة بمتغيرات نوعية أو كمية. البيانات هي حقائق خام غير منظمة تحتاج إلى معالجة.

المعلومات information

عندما تتم معالجة البيانات أو تنظيمها أو هيكلتها أو تقديمها في سياق معين لجعلها مفيدة، فإنها تسمى معلومات.

المجالات التي تستخدم فيها تكنولوجيا المعلومات:

- الأعمال و الصناعة
- البنوك
- التأمينات
- التسويق
- التعليم و السفر
- الترفيه و الفنون
- الطب

البتات والبايتات و الأرقام الثنائيّة

أولا: Bits and bytes

❖ كيف يمثل الحاسب البيانات؟

عندما ننظر إلى الحسب فإننا نرى نصوص، صور وأشكال.

بالنسبة للحاسب كل ذلك هو بيانات ثنائيّة أصفار ، واحدات.

السلسلة الآتية من الأصفار والواحدات تمثّل صورة

```
010001110100100101000110001110000011100101100001000000001
00000000000000010000000010000000000000000000000000000000000
0000000000000000111111111111111111111111110010000111111001
000001000000000010000000000000000000000000000000000101100
000000000000000000000000000000000000000000100000000000000001
0000000000000000000000010000000001010001000000000000000111011
```

السلسلة الآتية من الأصفار والواحدات تمثل أمر (تعليمة) لإضافة رقم

```
0001 0001 0000 0101
```

السؤال الذي يطرح نفسه هو لماذا تمثّل الحواسيب المعلومات بهذه الطريقة الصعبة؟ وكيف يمكن لسلسلة الأصفار والواحدات تمثيل عدّة أشياء مختلفة. هذا ما سنشرحه في هذه المحاضرة.

تستخدم الحواسيب البتات(الترقيمات الثنائيّة) لتمثيل البيانات كأصفار وواحدات. تعتبر البتات أصغر قطعة من المعلومات التي يستطيع الحاسب تخزينها.

❖ البتات (الترقميات الثنائيّة) Bits.

كلمة bit هي اختصار (binary digit)، وإنّ البت يخزّن إمّا القيمة صفر أو القيمة واحد، وبالتالي يمكن للبت تمثيل قيمتين مختلفتين(على الرغم من أنّ عدد القيم غير كبير وكنّه كافي لتمثيل أي حالة ثنائيّة) على سبيل المثال:

هل المصباح (**lightbulb**) مطفأ أو مضيء؟

State	Bit
	0
	1

هل الزر مفعّل أو غير مفعّل؟

State	Bit
Submit	0
Submit	1

هل الوقت الحالي هو قبل الظهر أو بعد الظهر؟

State	Bit
9:35 AM	0
9:35 PM	1

❖ سلسلة البتات Sequences of bits

يستخدم الحاسب عدّة بتات لتمثيل البيانات الأكثر تعقيداً التي لها عدّة قيم مقارنة مع البيانات البسيطة التي لها قيمتين فقط (On/Off).

سلسلة مكوّنة من بتين يمكنها تمثيل (2^2) أربعة قيم مختلفة: 00,01,10,11

سلسلة من ثلاثة بتات يمكنها تمثيل (2^3) ثمانية قيم مختلفة :

000, 001, 010, 011, 100, 101, 110, 111

يمكن أن تمثّل السلسلة عدّة أشياء: أرقام، محارف، بيكسل..... . بالإضافة لذلك يمكن لنفس السلسلة تمثيل عدّة أنماط مختلفة من البيانات على سبيل المثال: السلسلة 1000011 تمثل الرقم 67 في تطبيق حسابي كما أنّها تمثّل المحرف C في ملف نصي.
عادة تخزّن الحواسيب البتات باستخدام الترانزستورات الميكانيكية الكهربائية التي تربط الإشارات الكهربائية مع حالة On أو Off.

❖ البايتات Bytes

يعرف البايت على أنه وحدة المعلومات الرقمية التي تتكون من ثمانية بتات.
السلسلة الآتية تمثل بايت واحد 11110110،
والسلسة الآتية تمثّل ثلاثة بايتات 000010100101010011011011

❖ التحويل من بتات إلى بايتات

يتم التحويل من بتات إلى بايتات باستخدام عملية حسابية بسيطة: القسمة على 8 تحوّل من بت إلى بايت وبالمقابل الضرب ب 8 يحوّل من بايت إلى بت.
كم عدد البايتات في هذه السلسلة الثنائية الآتية:
1011100111011011011010010011010101111111
الجواب هو خمسة بايتات لأنه يوجد خمسة مجموعات من ثمانيات البتات أو بطريقة أخرى: إنّ طول السلسلة هم 40 بت فإذا قسمنا 40 على 8 فإن الناتج هو 5 وبالتالي إن هذه السلسلة تتكوّن من خمسة بايتات.

❖ لماذا البايت؟

ما هو الشيء الخاص حول 8 بت حتى تأخذ الاسم بايت؟ إنّ الحاسب يعالج كل البيانات على أنها بتات ولكنّه يفضّل التّعامل مع البتات بعد تجميعها في مجموعات ثمانية (بايتات) أو بمعنى آخر إنّ البايت هو كيف يفضّل الحاسب



ثانياً: Binary numbers

يوجد عدة أنواع للأنظمة الرقمية منها العشري (الذي يفهمه البشر) ومنها النظام الثنائي(الذي يفهمه الحاسب) والنظام الست عشري. في النظام العشري يتم العدّ إلى الرقم 10 كما يلي: 1-2-3-4-5-6-7-8-9-10، وكما مر معنا فإن الحاسب يمثّل كل المعلومات باستخدام البتات، فمن أجل تمثيل الأرقام بصيغة أصفار- واحدات فإنّ الحاسب يستخدم نظام الأرقام الثنائي حيث يتمّ العدّ للرقم 10 كما يلي: 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010

❖ الأرقام العشرية Decimal numbers:

في النظام العشري فإن الخانة الموجودة في أقصى اليمين هي مرتبة الآحاد تليها العشرات تليها المئات وهكذا، أي أن الرقم في الخانة الموجودة في أقصى اليمين مضروب بالرقم 1 والرقم التالي مضروب بالرقم 10، والرقم التالي مضروب بالرقم 100 وهكذا. على سبيل المثال يتم تمثيل الرقم 234 كما يلي:

2	3	4
hundreds' place	tens' place	ones' place
100	10	1

عند ضرب كل رقم بمنزلته فإننا نجد أنّ

$$234 = (2 * 100000 + (3 * 10) + (4 * 1))$$

يمكن تمثيل الرقم حسب قوة العدد عشرة، حيث يتم ضرب الرقم الموجود في خانة الآحاد ب (10^0) ويتم ضرب الرقم الموجود في خانة العشرات بالرقم (10^0)، ويضرب الرقم الموجود في المئات بالرقم (10^2).

2	3	4
hundreds' place	tens' place	ones' place
100	10	1
10^2	10^1	10^0

❖ الأرقام الثنائية Binary numbers

يعمل النظام الثنائي بنفس طريقة النظام العشري. الفرق الوحيد هو أنه بدلا من ضرب الأرقام بقوة العدد عشرة نضربهم بقوة العدد 2. لننظر إلى الرقم 1 يتم تمثيله في النظام الثنائي بالقيمة 0001.

0	0	0	1
8	4	2	1
2^3	2^2	2^1	2^0

وهذا يكافئ (1*1)+(0*2)+(0*4)+(0*8)

يتم تمثيل الرقم 10 في النظام الثنائي كما يلي: 1010

1	0	1	0
8	4	2	1
2^3	2^2	2^1	2^0

وهذا يكافئ (0*1)+(1*2)+(0*4)+(1*8)

❖ التحويل من النظام العشري إلى الثنائي

1- نرسم خط صغير من أجل كل بت. إذا كان الرقم أصغر من 16 نرسم 4 خطوط صغيرة، أما إذا كان أكبر من 16 وأصغر من 255 فإننا نرسم 8 خطوط صغيرة

2- تحت كل خط صغير نكتب قوة الرقم 2، حيث نبدأ من الخط الصغير في أقصى اليمين ونضع تحته 1 ونستمر بالضرب بالرقم 2 من أجل باقي الخطوط

3- نبدأ من الخط الصغير في أقصى اليسار و نختبر هل العدد المطلوب تحويله أكبر أو يساوي الرقم المكتوب تحت الخط الصغير، فإذا كان الجواب نعم فإننا نضع واحد على ذلك الخط الصغير، ونطرح تلك القيمة من العدد. أما إذا كان الجواب لا فإننا نضع صفر على ذلك الخط الصغير وننتقل إلى الخط الصغير التالي.

4- نكرر الخطوة 3 حتى يتم الحصول على القيمة 0 من عملية الطرح وعندها نكون قد انتهينا من تحويل العدد المطلوب إلى النظام الثنائي.

على سبيل المثال: تحويل العدد 6 من النظام العشري إلى الثنائي

6 أصغر من 16 لذلك نرسم أربع خطوط صغيرة وتحت كل خط صغير نضع قوة العدد 2 بدءا من أقصى اليمين كما يلي:

$$\frac{}{8}\;\frac{}{4}\;\frac{}{2}\;\frac{}{1}$$

لأن 6 أصغر من 8 فإننا نضع على الخط الصغير فوق 8 قيمة 0

$$\frac{0}{8}\;\frac{}{4}\;\frac{}{2}\;\frac{}{1}$$

لأن 6 أكبر من 4 فإننا نضع على الخط الصغير فوق 4 قيمة 1 ونطرح 4 من 6

$$\frac{0}{8}\;\frac{1}{4}\;\frac{}{2}\;\frac{}{1}$$

نتيجة طرح 4 من 6 يساوي 2 ولأنها لا تساوي الصفر فإننا نكرر الخطوة 3 أن العدد الذي ستتم مقارنته مع القيم المكتوبة تحت الخطوط الصغيرة هو الرقم 2 (ناتج الطرح)

$$\frac{0}{8}\ \frac{1}{4}\ \frac{}{2}\ \frac{}{1}\ \text{(Remainder: 2)}$$

2=2 لذلك نضع على الخط الصغير فوق 2 قيمة 1 ونطرح 2 من 2

$$\frac{0}{8}\ \frac{1}{4}\ \frac{1}{2}\ \frac{}{1}\ \text{(Remainder: 0)}$$

لأن نتيجة الطرح صفر فإن عملية التحويل تكون قد انتهت حيث نضع قيمة صفر على باقي الخطوط الصغيرة وهكذا ستكون نتيجة عملية التحويل

$$\frac{0}{8}\ \frac{1}{4}\ \frac{1}{2}\ \frac{0}{1}$$

ملاحظة: إذا نظرنا إلى الأرقام الآتية نلاحظ

Decimal	Binary
3	0011
5	0101
7	0111
9	1001

يجب أن يحتوي التحويل الثنائي للأرقام الفردية العشرية على 1 في البت الموجود في أقصى اليمين وهو بت مرتبة الآحاد أما الأرقام الزوجية تحتوي صفر في هذا البت.

بالنسبة للأرقام الفردية التي يتكون تمثيلها الثنائي من واحدات متتالية مثل 3-7-15 كما هو موضح في الشكل الآتي

Decimal	Binary
3	11
7	111
15	1111

فإنه توجد طريقة مباشرة للتحويل من النظام الثنائي إلى النظام العشري وهي أن نعد كم يوجد واحد في التمثيل الثنائي وليكن هذا العدد هو n فيكون الرقم العشري المقابل هو $2^n - 1$

على سبيل المثال إن الرقم العشري المقابل للتمثيل الثنائي الآتي: 11111 هو $2^5 - 1 = 31$

❖ مشكلة الطفحان overflow

عندما يخزن البرنامج الحاسوبي الأرقام ضمن متحولات فيجب على الحاسب أن يبحث عن طريقة لتمثيل تلك الأرقام في ذاكرة الحاسب. توجد سياسات مختلفة يمكن أن يتبعها الحاسب وذلك وفقا إذا كان الرقم هو رقم طبيعي أم لا. فوفقا لقيود الذاكرة يمكن أن تصادف البرامج الحاسوبية عدّة مشاكل منها: مشكلة الطفحان overflow.

الرقم الطبيعي هو أي رقم يمكن أن يكتب من دون استخدام الفاصلة مثل: 120- 10- 0- (20-) ...... وكما أصبح معروفا لنا فإن الحاسب يمثل هذه الأرقام باستخدام البتات أي كل رقم هو سلسلة من الأصفار والواحدات.

لتوضيح مشكلة الطفحان لنفترض أنه لدينا حاسب يستخدم أربع بتات لتمثيل الأرقام. حيث يستخدم البت الأول لتمثيل إشارة الرقم (هل هو رقم موجب أو سالب) ويستخدم البتات الثلاثة المتبقية من أجل قيمة الرقم. إذا سيتم تمثيل الرقم 1 كما يأتي:

0	0	0	1
+/-	4	2	1
sign	2^2	2^1	2^0

إذا كانت قيمة البت الذي يمثل إشارة الرقم هي صفر فإن الرقم موجب وإلا سيكون سالب.

أكبر رقم يمكن أن يمثله هذا الحاسب هو الرقم 7

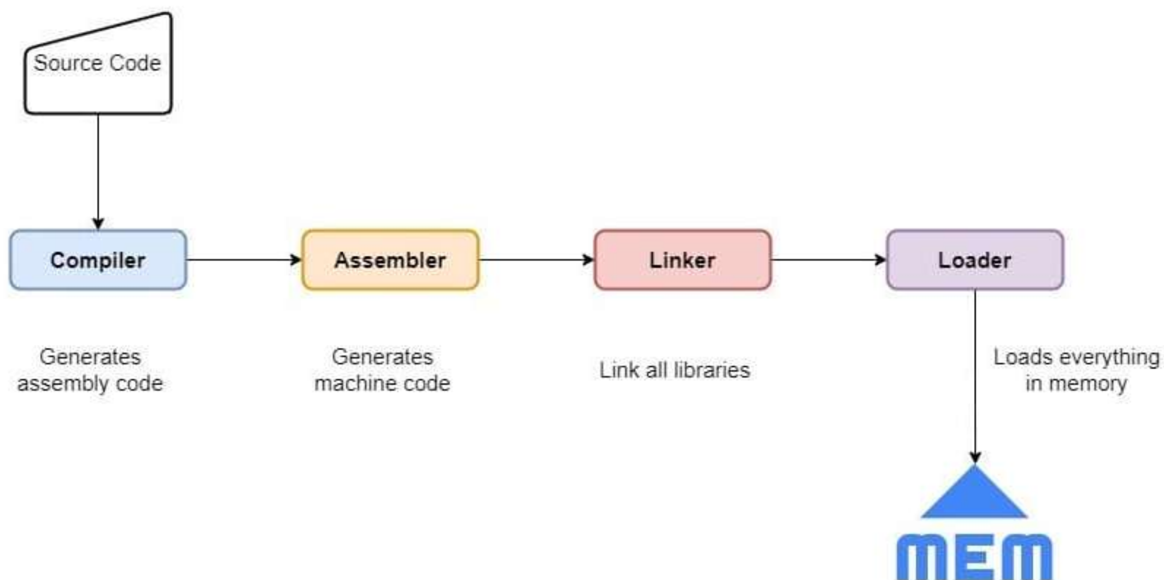
0	1	1	1
+/-	4	2	1
sign	2^2	2^1	2^0

ما لذي سيحدث إذا تم تشغيل البرنامج الآتي على هذا الحاسب؟

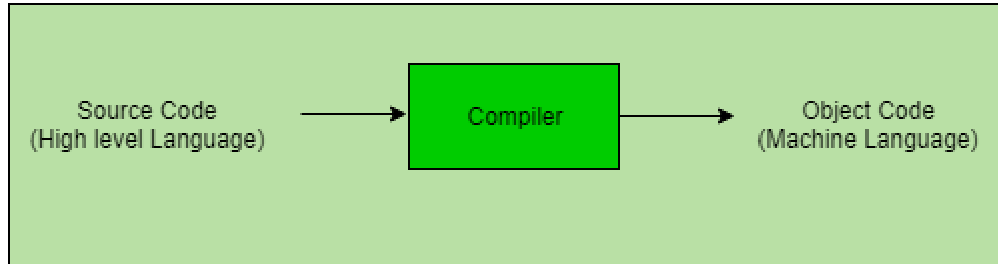
```
var x = 7;
var y = x + 1;
```

يخزن الحاسب المتحول x بشكل طبيعي، ولكن قيمة المتحول y أكبر من الرقم الأكبر الذي يستطيع الحاسب تخزينه. في هذه الحالة فإن الحاسب يعطي تقرير بحدوث مشكلة الطفحان " error overflow " أو أن يعرض الرسالة الآتية: " number is too large " أو أن يجعل كل الأرقام التي أكبر من 7 مساوية للرقم 7 أم أن يعمل تدوير للرقم الأكبر من 7 (أي أن تصبح قيمة الرقم 8 تساوي 1)

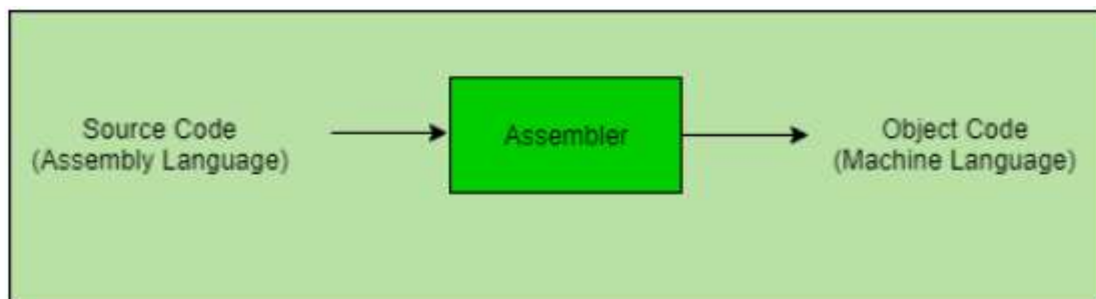
ثالثا: مفهوم المترجم و المجمع والمفسر



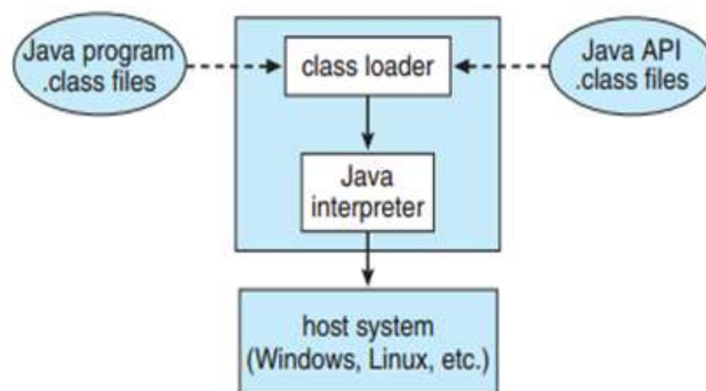
1- **المترجم compiler**: هو عبارة برنامج يقبل كدخل البرنامج المصدر المكتوب بلغة عالية المستوى ، وينتج برنامج مكتوب بلغة منخفضة المستوى (assembly language, byte code, object code, or machine code) دفعة واحدة.



2- **المجمع Assembler:** هو عبارة عن برنامج يأخذ كدخل الملف المكتوب بلغة منخفضة المستوى و الّذي صدر عن المترجم ويحوّله إلى ملف مكتوب بلغة الآلة والّذي يسمّى اصطلاحاً (object file)



3- **المفسر interpreter**: هو عبارة عن برنامج يحول البرنامج المكتوب بلغة عالية المستوى إلى برنامج مكتوب بلغة منخفضة المستوى حيث يقوم بعمله سطر تلو الآخر.



The Java virtual machine.

إعداد: د. أبي سلوم